Kolmogorov.ai | Continuity

Decision Ops

ООО Дата Сапиенс

v.1.1.0 | 1.11.2023

Table of contents

1. Руководство пользователя 3
1.1 О Continuity 3
2. Инструкция по установке 5
2.1 Введение 5
2.2 Администратору 6
3. Скачать документацию 19
3.1 Download 19

1. Руководство пользователя

1.1 О Continuity

1.1.1 Kolmogorov.ai Continuity

Continuity [ˌkɒntɪ'nju(ː)ɪti] - инструмент класса Decision Ops.

Программное обеспечение «Kolmogorov.ai Continuity» предназначено для анализа данных в бизнес-процессах различных индустрий: - финансовый сектор, - телеком, - ритейл, - производство.

Программное обеспечение является инструментом класса Decision Operations. Позволяет организовать работу с платформой для данных и аналитики как с единым целым, а также выстроить связь между процессами анализа данных и контекстом решаемой бизнес-задачи.

Программное обеспечение «Kolmogorov.ai Continuity» поддерживает следующие функции: - Регистрация проекта бизнес-процесса; - Исполнение проекта бизнес-процесса; - Создание шаблона этапа бизнес-процесса; - Создание шаблона проекта бизнес-процесса.

1.1.2 Глоссарий

Объект (Object) - объект системы Continuity представляет собой некоторую сущность. Объектом являются проект, этап, задача, артефакт, а также шаблоны проекта, этапа, задачи и артефакта.

Объектная модель Continuity (Object model) - это иерархически устроенная структура объектов системы, где вышестоящие сущности объединяют в себе нижестоящие с помощью нод. Так артефакты являются сущностями внутри задачи, набор задач содержится в этапе, этапы объединяются проектом.

Нода (Node) - это сущность системы. Имеет двойное значение: 1. Нода представляет собой контейнер для упаковки объекта. 2. Нода является узлом графа continuity.

Проект (Project) - это верхнеуровневый объект системы, объединяющий этапы с объектами внутри них. Проект может быть заведен для различных целей, он объединяет сущности с общей бизнес-задачей.

Этап (Stage) - это шаг workflow проекта. Он содержит ряд логически объединенных задач, которые выполняются параллельно или последовательно в соответствии с автоматизированным BPMN процессом.

Задача (Task) - единица бизнес-процесса, предназначенная для выполнения какого-то действия пользователем или сервисом.

Артефакт (Artefact) - это некоторый атрибут бизнес-процесса, который содержит необходимые на соответствующем этапе информацию/ссылку/файл. Артефакты добавляются пользователем в ходе выполнения задач.

Граф Continuity (Graph) - это workflow проекта, имеющий форму графа. Он содержит в себе ноды и ребра. В нодах содержатся этапы проекта, а ребра определяют направление workflow.

Ребро графа (Edge) обеспечивает связь между нодами. По ребрам графа двигается workflow проекта.

Шаблон объекта (Template) - это описание объекта и его свойств. По умолчанию запуск проекта и создание объектов происходит в соответствии с шаблоном. Это необходимо в случае переиспользования workflow/этапа/задачи более одного раза. В шаблоне может быть описано, из каких этапов и задач состоит workflow, информация об артефактах, а также свойства объектов.
Шаблон имеет вложенную структуру: каждый объект содержит набор нод-контейнеров, в каждом из которых содержится объект.
То есть шаблон проекта содержит список нод с шаблонами этапов, шаблон этапа - список нод с шабдлонами задач, шаблон задачи - список нод с шаблонами артефактов.

Контекст объекта (Context) - свойство объекта. С помощью контекстов можно настраивать объекты в рамках шаблона.

Свимлейн (Swimlane) - это инструмент Continuity для настройки доступов к объектам.

Ключ (Key) - это бизнес-идентификатор (то есть текстовый ключ) объекта в системе.

Тэг (Tag) - это название версии объекта.

Бизнес тэг (Business tag) - это метка, которая может быть добавлена к любому объекту системы для использования при фильтрации.

Continuity SDK - python-библиотека предоставляющая полный функционал для удобной работы с API в интерфейсе Jupyter.

2. Инструкция по установке

2.1 Введение

Continuity обеспечивает поддержку жизненного цикла систем анализа данных в бизнес-процессах различных индустрий.

Основной сервис бэкенда - FastAPI сервер, который принимает запросы по адресу, описанному в ingress сервиса. Часть задач выполняется асинхронно с использованием Celery + Redis. Данные хранятся в БД Postgres, а также хранилище S3.

Взаимодействие пользователя с приложением осуществляется через web-интерфейс. Доступ к функционалу открывается после авторизации.

Установка

Первичная установка программного обеспечения производится командой разработки и внедрения ООО «Дата Сапиенс» на сервер, предоставляемый заказчиком.

Обновление

Проверка, загрузка и установка обновлений программного обеспечения требует подключения к Интернету и выполняется вручную на сервере администратором ПО. Для обновления комонента ПО необходимо удалить его и установить повторно, после чего снова запустить Kolmogorov.ai **Continuity**.

Переустановка

Любую доступную версию Kolmogorov.ai **Continuity** можно загрузить и установить с помощью контейнеров, поставляемых разработчиком. Перед переустановкой допускается удаление существующего ПО.

2.2 Администратору

2.2.1 Пререквизиты

Компоненты, которые не входят в дефолтную поставку приложения Continuity, однако необходимы для его работы.

Kubernetes/OpenShift

`K8s`-совместимый кластер с возможностью заведения отдельного namespace. В Данном namespace - возможность создания сервисной УЗ с правами на CRUD всех базовых типов объектов. Минимальная ресурсная квота - `8CPU, 16Gb RAM, 100Gb disk space`. Также, на локальной машине, должен быть установлен инструмент для управления Kubernetes приложениями - **Helm**. Дополнительно в кластере понадобятся:

- `NFS StorageClass` - RWM класс хранения в Kubernetes, который использует NFS-сервер.
- `Ingress Controller` - контроллер, который обеспечивает доступ к сервисам Kubernetes извне.

Установка Kubernetes на minikube

Keycloak

Сервис для авторизации и аутентификации **Keycloak**.

Рекомендуется использовать чарт codecentric. Описание установки и настройки сервиса можно найти здесь.

S3 хранилище

S3 хранилище (Minio или Yandex Object Storage) - хранилище, в котором сохраняются результаты метрик. В данном хранилище должна быть возможностью заведения отдельного бакета, минимально рекомендованный размер которого - `100Gb`. Для работы с данным бакетом нужен сервисный аккаунт с ролью `storage.editor` или `storage.admin` (GET/UPDATE/DELETE объектов).

Инструкция по установке Minio

2.2.2 Установка Continuity в Kubernetes

Доступ к образам и чартам

> **Важно**
>
> Необходимые учётные данные для доступа к репозиториям можно получить по запросу.

ПРИМЕР ПОЛУЧЕНИЯ DOCKER ОБРАЗА ИЗ РЕПОЗИТОРИЯ:

```
export USERNAME=
export PASSWORD=
export URL_DOCKER_REGISTRY=
export IMAGE=
docker login $URL_DOCKER_REGISTRY --username $USERNAME --password $PASSWORD
docker pull $IMAGE
```

ПРИМЕР ПОЛУЧЕНИЯ HELM ЧАРТА ИЗ РЕПОЗИТОРИЯ:

```
export URL_CHART_REGISTRY=
export CHART_NAME=
export CHART_VERSION=
helm repo add continuity $URL_CHART_REGISTRY --username $USERNAME --password $PASSWORD
helm repo update
helm search repo $CHART_NAME -l --devel
helm pull continuity --version $CHART_VERSION
```

Continuity

УСТРОЙСТВО ЧАРТА

Чарт содержит в себе следующие сервисы:

`continuity-backend`

Основной сервис бэкенда. Внутри контейнера разворачивается FastAPI сервер. Также сервис содержит два init-container:

- `init-db` - применяющий миграции Alembic к базе данных;

`continuity-celery`

Celery сервис, который отвечает за изменение состояния графов сущностей на continuity (задачи синхронизации). Сервис делает запросы к базе, делает в ней изменения и рассчитывает состояния графов.

`continuity-postgresql`

Сервис, который отвечает за хранение базы данных continuity.

`continuity-redis`

Сервис, который отвечает за хранение очереди задач для Celery.

ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ

Название	Описание	Обязательная	Значение по умолчанию
CONTINUITY_ROOT_PATH	Базовый путь до API Continuity `(пример: "/api")`	Да	""
CONTINUITY_ADMIN_ROLE	Название роли администратора	Нет	"continuity_admin"
CONTINUITY_AUDIT_ROLE	Название роли аудитора	Нет	"continuity_audit"
CONTINUITY_LOG_READER_ROLE	Название роли с доступом к логам	Нет	"continuity_log_reader"
CONTINUITY_DB_URL	URL базы Данных для Continuity (пример: "postgresql:// postgres:postgres@continuity-postgresql:5432/postgres")	Да	
CONTINUITY_DB_POOL_SIZE	Максимальное количество открытых соединений с базой	Нет	30
CONTINUITY_DB_CONNECT_TIMEOUT	Время таймаут соединения с базой	Нет	10
CONTINUITY_CHILD_KEY	Форма генерации имени runtime объектов	Нет	"{parent.FCT_ENTITY_KEY}-{parent.FCT_ENTITY_VERSIO {template.DIM_ENTITY_KEY}
CONTINUITY_CELERY_NAME	Имя для процесса Celery воркера	Нет	"continuity"
CONTINUITY_CELERY_BROKER_URL	URL брокера для Celery воркера	Да	
CONTINUITY_CELERY_RESULT_BACKEND	Бэкенд результатов для Celery воркера	Да	
CONTINUITY_S3_URL	URL s3 для Continuity (пример: "https:// storage.yandexcloud.net")	Да	
CONTINUITY_S3_ACCESS_KEY	Ключ доступа S3 для Continuity (пример: "AccessKey")	Да	
CONTINUITY_S3_SECRET_ACCESS_KEY	Секретный ключ доступа S3 для Continuity (пример: "AccessSecret")	Да	
CONTINUITY_S3_BUCKET	Имя бакета S3 для Continuity (пример: "continuity")	Да	
CONTINUITY_S3_PREFIX	Префикс с именем папки S3 для Continuity	Нет	"continuity"
CONTINUITY_S3_VERIFY	Проверять ли SSL-сертификаты S3 (пример: "False")	Нет	"False"
CONTINUITY_K8S_NAMESPACE	K8s namespace	Нет	"continuity"

Название	Описание	Обязательная	Значение по умолчанию
KEYCLOAK_VERIFY	Проверять ли SSL-сертификаты KEYCLOAK (пример: "False")	Нет	"False"
KEYCLOAK_URL	URL для аутентификации Keycloak (пример: "https://auth.k8s.datasapience.ru/auth")	Да	
KEYCLOAK_REALM	Realm для Keycloak (пример: "dev")	Да	
KEYCLOAK_CLIENT_ID	ID клиента для Keycloak (пример: "kolmogorov")	Да	
KEYCLOAK_CLIENT_SECRET	Секрет	Нет	"client-secret"
KEYCLOAK_ADMIN_USERNAME	Имя пользователя администратора для Keycloak (пример: "writer")	Да	
KEYCLOAK_ADMIN_PASSWORD	Пароль администратора для Keycloak (пример: "writer")	Да	
CONTINUITY_SENTRY_ENABLED	Включен ли Sentry для Continuity (пример: "False")	Нет	"False"
CONTINUITY_SENTRY_DSN	DSN Sentry для Continuity (пример: "")	Нет	

УСТАНОВКА/ОБНОВЛЕНИЕ/УДАЛЕНИЕ РЕЛИЗА CONTINUITY

Установка/обновление

1. Получите дефолтный `values-continuity.yaml` из чарта:

```
export CHART_NAME=
export CHART_VERSION=
helm show values CHART_NAME --version $CHART_VERSION >> values-continuity.yaml
```

2. Отредактируйте файл `values-continuity.yaml`:
3. Укажите актуальный репозиторий образа и тег образа в `global.image.repository` и `global.image.tag`
4. Подставьте актуальные данные для ingress и в следующих сервисах

- `service.ingress.host`, `service.ingress.baseDomain`

5. Укажите актуальные переменные окружения `service.secrets.data`, которые описаны в разделе Переменные окружения; минимально необходимо указать следующие переменные:

- `KEYCLOAK_URL`
- `KEYCLOAK_REALM`
- `KEYCLOAK_CLIENT_ID`
- `KEYCLOAK_ADMIN_USERNAME`
- `KEYCLOAK_ADMIN_PASSWORD`
- `CONTINUITY_S3_URL`
- `CONTINUITY_S3_ACCESS_KEY`
- `CONTINUITY_S3_SECRET_ACCESS_KEY`
- `CONTINUITY_S3_BUCKET`

6. Запустите команду для установки релиза

```
helm upgrade
--namespace=<namespace> \
--install \
--atomic \
--values values-continuity.yaml \
<realisename> <path to chart>
```

где `namespace` - неймспейс k8s, в котором устанавливается релиз, `realisename` - название релиза, `path to chart` - путь до папки чарта или до архива

Удаление

Для удаления чарта запустите команду:

```
helm delete --namespace=<namespace> <realisename>
```

где `namespace` - неймспейс k8s, в котором устанавливается релиз, `realisename` - название релиза,

> **Важно**
>
> Для полного удаления релиза необходимо: 1. Удалить секреты, с именем всех существующих источников данных; 2. Удалить папку в бакете, указанную в переменной окружения `Continuity_S3_PREFIX`. 3. Запустить команду `helm delete --namespace=<namespace> <realisename> --purge`.

Continuity UI

УСТРОЙСТВО ЧАРТА

Чарт содержит в себе один сервис:

`continuity-ui`

Сервис с файлами web-приложения. Также сервис содержит init-контейнер:

- `generate-configs` - настраивает переменные окружения и создает файл конфигурации.

ПЕРЕМЕННЫЕ ОКРУЖЕНИЯ

Название	Описание	Обязательная
CONTINUITY_API	Базовый путь до API Continuity `(пример: "/api")`	Да
TITLE	Заголовок приложения, отображаемый в интерфейсе логирования `(пример: "Predicate")`	Да
SUBTITLE	Подзаголовок приложения, отображаемый в интерфейсе логирования `(пример: "Metric Store - валидация ML Моделей")`	Да
KEYCLOAK_URL	Адрес Keycloak для авторизации, обычно совпадает с указанным в бекенде `(пример: "https://<keycloak-url>/auth")`	Да
KEYCLOAK_REALM	Название realm в Keycloak `(пример: "dev")`	Да
KEYCLOAK_CLIENT_ID	Идентификатор клиента в Keycloak `(пример: "kolmogorov")`	Да
KEYCLOAK_CLIENT_SECRET	Секретный ключ клиента Keycloak, по умолчанию пустой `(пример: "")`	Да
TZ	Часовой пояс `(пример: "Europe/Moscow")`	Да
SENTRY_ENABLED	Флаг, указывающий, включен ли сервис Sentry для отслеживания ошибок `(пример: "true")`	Да
SENTRY_DNS	DNS-адрес проекта Sentry `(пример: "https://<url-sentry-project>/<project_number>")`	Да

УСТАНОВКА/ОБНОВЛЕНИЕ/УДАЛЕНИЕ РЕЛИЗА CONTINUITY UI

Установка/обновление

1. Получите дефолтный `values-continuity-ui.yaml` из чарта:

```
export CHART_NAME=
export CHART_VERSION=
helm show values CHART_NAME --version $CHART_VERSION >> values-continuity-ui.yaml
```

2. Отредактируйте файл `values-continuity-ui.yaml`

- Укажите актуальный репозиторий образа - `service.image.repository` и тег образа - `service.image.tag`;
- Укажите актуальный `service.ingress.host` и `service.ingress.baseDomain`;
- Укажите актуальные переменные окружения `service.secrets.data`, которые описаны в разделе Переменные окружения; минимально необходимо указать следующие переменные:
- `KEYCLOAK_URL`
- `KEYCLOAK_REALM`
- `KEYCLOAK_CLIENT_ID`

3. Запустите команду

```
helm upgrade \
--namespace=<namespace> \
--install \
--atomic \
--values values-continuity-ui.yaml \
<realisename> <path to chart>
```

где `namespace` - неймспейс k8s, в котором устанавливается релиз, `realisename` - название релиза, `path to chart` - путь до папки чарта или до архива

Удаление

Для удаления чарта запустите команду:

```
helm delete --namespace=<namespace> <realisename>
```

где `namespace` - неймспейс k8s, в котором устанавливается релиз, `realisename` - название релиза,

> **Важно**
>
> Для полного удаления релиза необходимо запустить команду `helm delete --namespace=<namespace> <realisename> --purge`.

2.2.3 Установка и настройка Keycloak

УСТАНОВКА KEYCLOAK

> **Важно**
>
> Если у вас уже есть Keycloak, то пропустите этот шаг и перейдите к его настройке.

Для установки Keycloak можете воспользоваться следующим чартом

Перед установкой необходимо создать следующий файл `values.yaml`, подставив везде свое значение в переменную `IP_LOADBALANCER`. Если у вас есть собственное доменное имя, то замените `IP_LOADBALANCER.nip.io` на свое доменное имя.

Файл `values.yaml`

```yaml
keycloakUser: "admin"
keycloakPassword: "admin"


databaseUser: "keycloak-user"
databasePassword: "dbpassword"

replicas: 1


extraEnv: |
 - name: JAVA_OPTS
   value: >-
     -XX:+UseContainerSupport
     -XX:MaxRAMPercentage=50.0
     -Djava.net.preferIPv4Stack=true
     -Djboss.modules.system.pkgs=$JBOSS_MODULES_SYSTEM_PKGS
     -Djava.awt.headless=true
     -Dkeycloak.profile.feature.upload_scripts=enabled
 - name: KEYCLOAK_LOGLEVEL
   value: INFO
 - name: PROXY_ADDRESS_FORWARDING
   value: "true"


extraEnvFrom: |
 - secretRef:
     name: '{{ include "keycloak.fullname" . }}-cred'
 - secretRef:
     name: '{{ include "keycloak.fullname" . }}-db'

secrets:
 cred:
   stringData:
     KEYCLOAK_USER: "{{ .Values.keycloakUser }}"
     KEYCLOAK_PASSWORD: "{{ .Values.keycloakPassword }}"
 db:
   stringData:
     DB_USER: '{{ .Values.databaseUser }}'
     DB_PASSWORD: '{{ .Values.databasePassword }}'


# resources:
#   requests:
#     cpu: 500m
#     memory: 1024Mi
#   limits:
#     cpu: 2000m
#     memory: 2048Mi


ingress:
 enabled: true
 ingressClassName: "nginx"
 servicePort: http
 annotations:
   ingress.kubernetes.io/ssl-redirect: "true"
   nginx.ingress.kubernetes.io/ssl-redirect: "true"
   nginx.ingress.kubernetes.io/proxy-body-size: "128k"
   nginx.ingress.kubernetes.io/server-snippet: |
     more_set_headers "Access-Control-Allow-Origin: $http_origin";
     location ~* /auth/realms/[^/]+/metrics {
         return 403;
     }
 rules:
   - host: "IP_LOADBALANCER.nip.io"
     paths:
       - path: /auth
         pathType: Prefix
 tls:
   - hosts:
       - "IP_LOADBALANCER.nip.io"
     secretName: dev-wildcard
 console:
   enabled: true
   ingressClassName: "nginx"
   annotations:
     ingress.kubernetes.io/ssl-redirect: "true"
     nginx.ingress.kubernetes.io/ssl-redirect: "true"
     nginx.ingress.kubernetes.io/proxy-body-size: "128k"
   rules:
     - host: "IP_LOADBALANCER.nip.io"
       paths:
         - path: /auth/admin/
           pathType: Prefix
   tls:
     - hosts:
         - "IP_LOADBALANCER.nip.io"
       secretName: dev-wildcard
```

После создания файла, выполните установку Keycloak:

```
helm repo add codecentric https://codecentric.github.io/helm-charts
helm repo update
helm upgrade --install keycloak codecentric/keycloak \
--values values.yaml \
--namespace default \
--kube-context $CONTEXT_NAME
```

НАСТРОЙКА ДОСТУПА К KEYCLOAK

1. Создайте клиента под именем `klmg` и добавить клиенту в поле Valid redirect URIs текущий host в формате `https://{HOST}/*` (для локальной установки - `https://IP_LOADBALANCER.nip.io/*`).
2. В настройках клиента поле `Access Type` должно быть в значении `public`. В более новых версиях Keycloak, аналогом поля `Access Type` являются поля `Client authentication` и `Authorization`, они должны быть в состоянии `off`.
3. `В дефолтной версии:`

   В настройках клиента необходимо перейти в `Mappers`, нажать кнопку `Create`.

   `В новой версии:`

   В настройках клиента необходимо перейти в `Client scopes`. Из дефолтного списка `Assigned client scope` выбрать `<название клиента>-dedicated` и добавить маппер `by configuration` нажав `Add mapper`.

   В настройках указать:

   - Name: roles
   - Mapper type: User Client Role
   - Token Claim Name: roles

   Остальные настройки оставить по умолчанию, нажать `Save`.

1. В настройках клиента необходимо перейти в `Roles` и создать роль `continuity_admin` для предоставления выбранным пользователям прав админа (возможность видеть и изменять все сущности созданные на стенде любым пользователем).
2. Создать необходимых пользователей и выдать им при необходимости роль `continuity_admin`. Для этого перейдите в раздел `Users`, выберите пользователя, перейдите в раздел `Role Mappings`.

   `В дефолтной версии:`

   В поле `Client Roles` выберите клиента `klmg`, в поле `Available Roles выберите роль` выберите `continuity_admin` и нажмите `Add selected`.

   `В новой версии:`

   Нажмите `Assign role` -> `Filter by client`. В поиске введите `continuity_admin`, выберите роль и нажмите `Assign`.

3. Скачать документацию

3.1 Download

- Руководство пользователя [PDF]
- Руководство по установке [PDF]
- Функциональные характеристики [PDF]
- Жизненный цикл [PDF]